Data Mining With MicroArray Explorer - A Java-based Open Source Tool

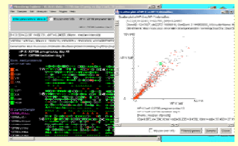
Peter F. Lemkin, Ph.D.

LECB, NCI-Frederick

http://maexplorer.sourceforge.net/
or
http://www.lecb.ncifcrf.gov/MAExplorer/

NIH BCIG/BITS (Nov 13, 2002)
Revised: 10-30-2002

Outline



I. **Introduction to Data Mining**

II. Overview of MAExplorer

III. Example

IV. Overview of Java Plug-ins for MAExplorer

V. Overview of Open Source Development

Microarrays for Studying mRNA Expression

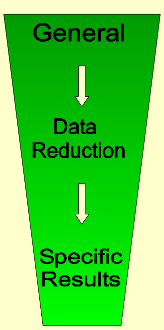
- Traditionally, biochemical and genetic pathways have been studied on a gene by gene basis
- cDNA and oligonucleotide microarrays profile >30,000 genes that can be studied as a function of experimental conditions
- Clonal DNA or oligonucleotides are attached to array substrates
- mRNA from an experiment is reverse transcribed to labeled cDNA (^{33}P, or Cy3/Cy5 dyes) or cRNA (biotin) and hybridized to an array
- Hybridized arrays are scanned and images are quantified
- Bioinformatic tools are required to handle large amount of data

Schematic Overview of cDNA & Oligo Arrays

A. Schultz & J. Downward (2001) Navigating gene expression using microarrays — a technology review. Nature Cell Biology 3: E190-E195.

See Figure 1 Schematic overview of probe array and target preparation for spotted cDNA microarrays and high-density oligonucleotide microarrays.

Array Data Mining - Finding Patterns of Genes



Quantified array spot data for multiple samples and replicates

Organize by: sample, gene expression, gene sets

Change views: normalization, data filters

Visualize and query: plots, cluster, reports

Explore: external genomic databases

Subset of genes for further analysis

The Problem

- Conceptualize data as a “spreadsheet” of samples vs. normalized quantified microarray gene expression data
- **What do we do with all those spots?**
- Could look for **putative patterns of changes** of experimental conditions with gene expression data
- **Correlation of gene expression changes** with biological state implies a relationship but does not imply cause and effect
- Results are a **starting point for further analysis** using other tools - PubMed, genomic databases, other lab experiments, etc.

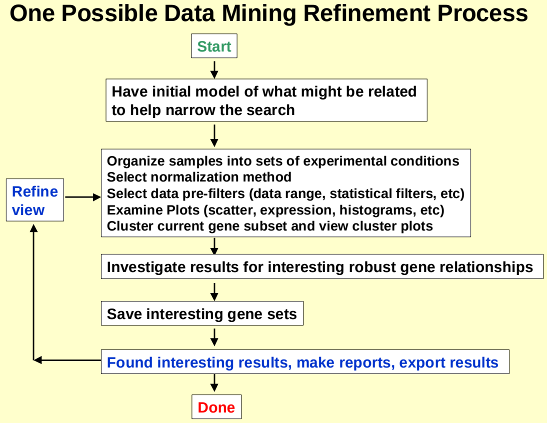
Things To Consider in Data Mining

- Initially, **don't know what patterns to look for**
- A good **experimental design** will enable performing data analysis where changes might be expected based on the biology, the number of samples, genes, and noise in the system
- **Look for the differences** between resulting expression patterns
- Resulting patterns along with knowledge of the underlying biology **may help form hypotheses** that could be further tested

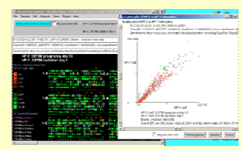
Data Mining is a Process

- ***Data mining is a <u>pattern discovery activity</u> - use all the tools available***
- It is **open-ended** because of the variety of ways data may be partitioned, normalized, pre-filtered, clustered, and viewed
- How do these tools help **find patterns**?
- By visual, statistical, and clustering methods
- Be careful to take the false discovery rate into account

One Possible Data Mining Refinement Process



Outline



I. Introduction to Data Mining

II. **Overview of MAExplorer**

III. Example

IV. Overview of Java Plug-ins for MAExplorer

V. Overview of Open Source Development

What is MicroArray Explorer?

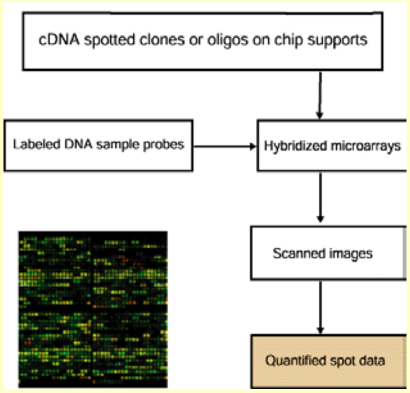
MAExplorer is an Open Source Java-based microarray data mining tool

- Initially developed for Mammary Genome Anatomy Program (MGAP) - Hennighausen et al. (NIDDK)
- Lemkin et al. *Nucleic Acids Res.* (2000) **28**:4452
- Handles multiple cDNA or oligo arrays, replicate spots
- Handles intensity or ratio (Cy3/Cy5) quantified array data

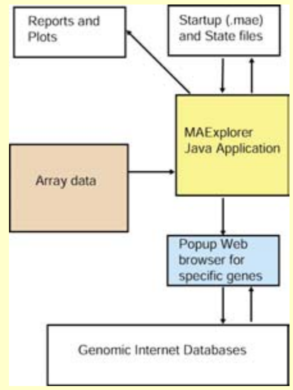
What is MicroArray Explorer? (continued)

- Analyzes data for 2- and N-condition expression profiles
- Data filters create gene sets using statistics, clustering, other gene sets
- Allows direct manipulation of data in graphics and spreadsheets
- Accesses genomic Web servers from plots and reports
- May be extended by writing Java plug-ins
- Available as Open Source at SourceForge.net

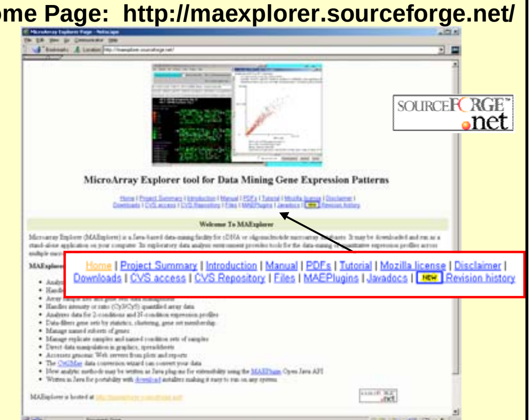
MAExplorer - Data Preparation



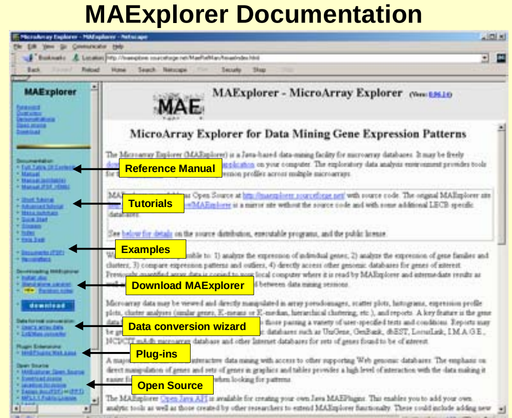
Paradigm: Local & Genomic-Web Databases



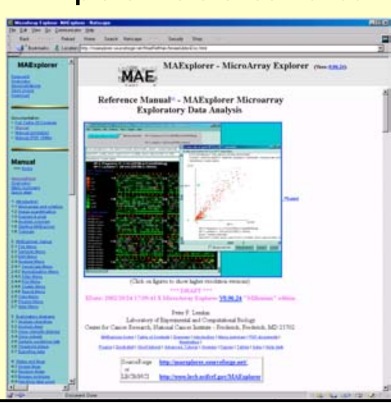
Home Page: http://maexplorer.sourceforge.net/



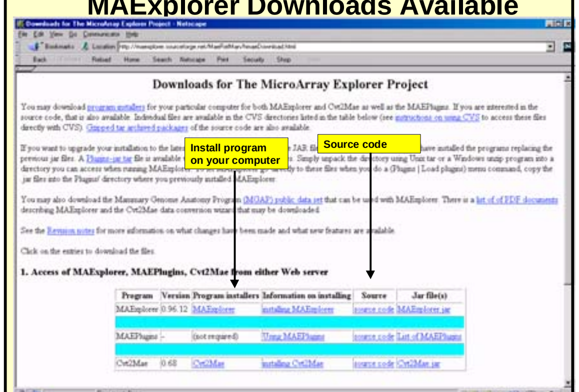
MAExplorer Documentation



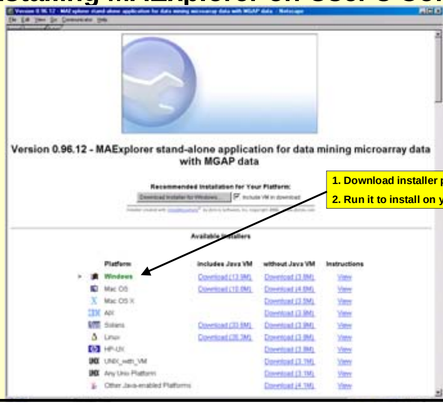
MAExplorer Reference Manual



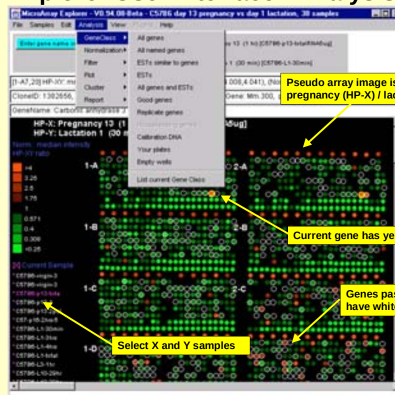
MAExplorer Downloads Available



Installing MAExplorer on User's Computer

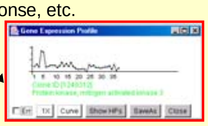


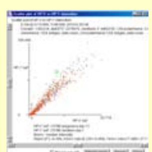
MAExplorer User Interface - Analysis Menu



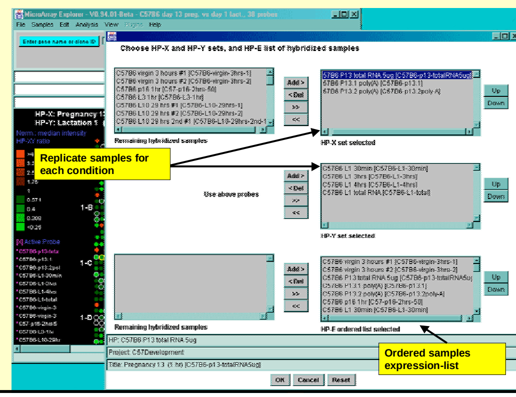
Split Samples into 2-Condition Sets and Ordered N-Condition Lists

- The **2-class division allows using sets of replicate samples** for computing better gene expression estimates and allows using *t*-Tests etc. to determine statistical significance
- The **ordered of N samples** is used to represent an **expression profile** list. E.g., time-series, development stages, drug-dose response, etc.





Sample Conditions (X vs Y) Sets & Expression Lists



Quantified Data Used in Microarray Analysis

- 1) **Sets of samples** using either intensity (^{33}P radio- or biotin-labeled) or ratio (Cy5/Cy3 fluorescent-labeled) DNA
- 2) Each **hybridized sample** contains thousands of spots correlated to spotted clones or oligonucleotides
- If **^{33}P or biotin,** then normalize data **between** hybridized array samples by large numbers of common "housekeeping" genes
- If **(Cy3, Cy5),** then use either Cy3 or Cy5 as a standard reference sample **within** a sample to normalize **between** samples
- Must normalize data between samples to be able to compare them

Normalize Intensity Data (^{33}P, Biotin) Between Samples

- Assuming linearity, for each array sample **j** get an estimate T_j of total spot labeling for a common subset of genes
- **Methods for estimating T_j** : mean, median, log median, Zscore, log Zscore, sum of calibration DNA, sum of gene set, etc.
- **Compute T_j over specific gene set**: calibration genes, all genes on the array, specific subset of genes
- Scale spot data **within** each sample (for samples 1 and 2, gene **k**):

$$s^*_{1,k} = s_{1,k} / T_1$$

and

$$s^*_{2,k} = s_{2,k} / T_2$$

- Then we may **compare** normalized $s^*_{1,k}$ **and** $s^*_{2,k}$ values

Normalize Ratio Data (Cy3, Cy5) Between Samples

- Let Cy3-labeled spots be the common **reference sample** hybridized to all arrays. Then independent samples are labeled with Cy5
- **Within** each sample, Cy5 data is scaled by corresponding common reference Cy3 spot values (samples 1 and 2, and all genes k) to compute ratio values (s^r_{1k}, s^r_{2k}) :

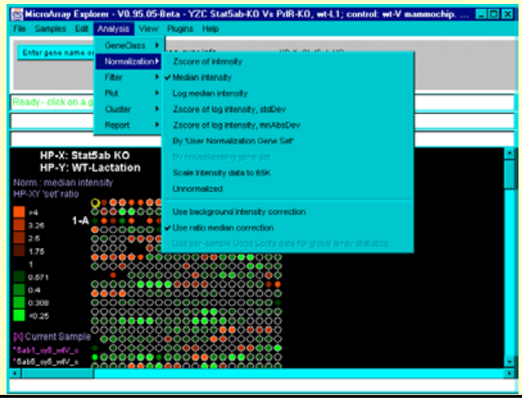
$$s^r_{1k} = s_{1k,cy5} / s_{1k,cy3}$$

and

$$s^r_{2k} = s_{2k,cy5} / s_{2k,cy3}$$

- Then **compare** the normalized s^r_{1k} and s^r_{2k} values
- Also commonly used non-linear methods such as loess, quantile

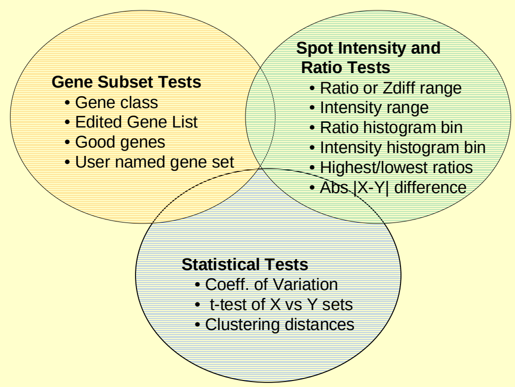
Normalization Menu



Multiple Data Filters - Finding a Gene Subset

- A data filter is applied to all genes for filter tests selected
- Creates a working subset of genes used for subsequent clustering, plots, and reports
- Data filter computes intersection (AND) of sets:
  1. Gene subsets defined in previous operations
  2. Spot intensity & ratio ranges, quality & detection
  3. Statistics: CV, t-Test
  4. Clustering: similar expression profiles, K-means, hierarchical - dendrograms & clustergrams
- Changing filter parameters recomputes data filter and then active plots or clustering methods

Gene Set Data Filter is Intersection of Tests



Example: the problem with fold change

- A measure of difference between 2 samples is "fold change"
  f(x,y)= x/y
- However ***f*** is sensitive to noise. If noise in all measurements is constant ***e***, then $f_e(x,y,e)$ has a range of values
  [*(x-e)/(y+e)* to *(x+e)/(y-e)*]
- ***Example:*** for two points ***(x,y)*** = (6,3) & (600,300), and ***e*** = 0.5 then the range of fold change for these two points is

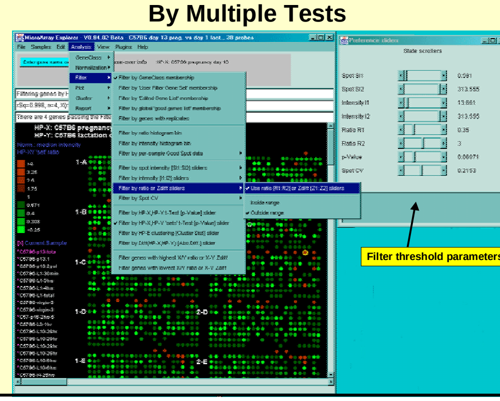
$f(6,3)$ = 2.0
$f_e(6,3,.5)$ = [5.5 / 3.5 to 6.5 / 2.5] = [**1.57 to 2.6**],

and

$f(600,300)$ = 2.0
$f_e(600,300,.5)$ = [559.5 / 300.5 to 600.5 / 299.5]
= [**1.995 to 2.005**].

[from I. Kohane, Apr, 2001]

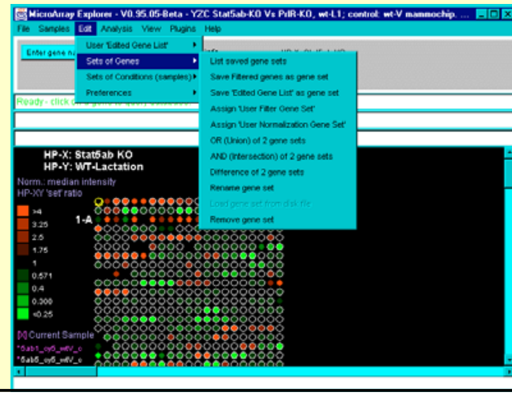
Filter Menu - Gene Data Operations By Multiple Tests



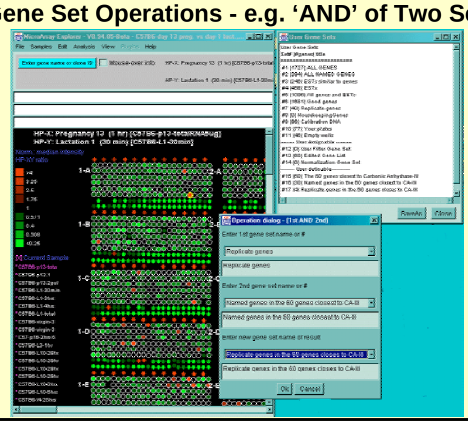
Data Management Tools - Named Gene and Sample Condition Sets

- Named gene sets - save/edit data filter/cluster results
- Named condition sets - save/edit subsets of samples
- Perform Boolean operations (AND, OR, DIFFERENCE) on sets to create new sets
- Use sets in subsequent data filters, normalization, analyses, plots, reports, and data management

Edit Menu - Gene Sets & Condition Lists



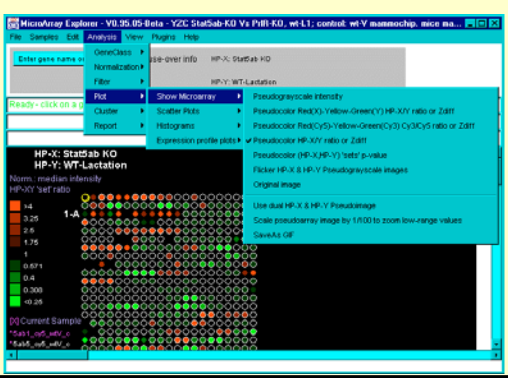
Gene Set Operations - e.g. 'AND' of Two Sets



Graphic Plots

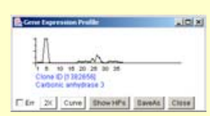
- Pseudoarray images - intensity, ratios of conditions, ratios of averaged conditions, p-values, etc.
- Scatter plots - channel vs. channel, sample vs. sample, mean conditions vs. mean conditions, similar genes, K-means clusters
- Histograms - sample ratios, intensity
- Expression profiles - individual genes or sets of genes
- Clustering displays - silhouette similarity plots, K-means clusters, clustergrams (heat maps), dendrograms

Plot Menus - e.g. Pseudoarray Images of Functions of Samples

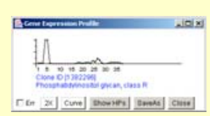


Which Genes Are Most Similar? Compare Gene Expression Profiles

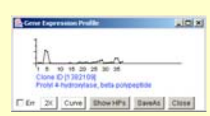
A)



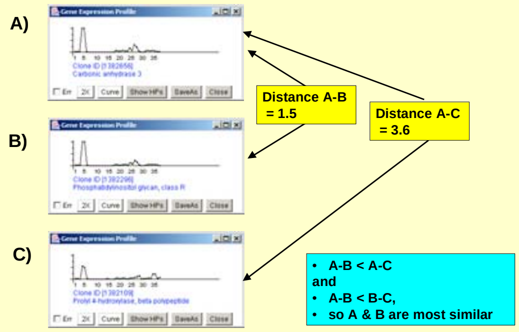
B)



C)



Which Genes are Most Similar? Compare Gene Expression Profile Distances



Definition: Gene Expression Profile

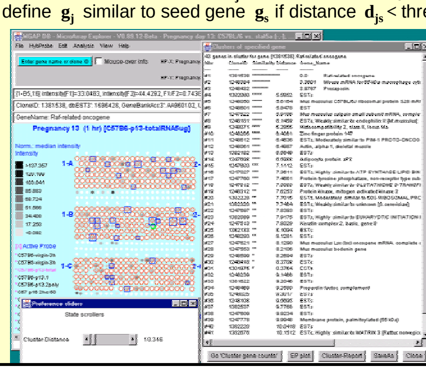
- An **expression profile** $\mathbf{e_j}$ is an ordered list of $\mathbf{N}$ normalized spot values of samples $\mathbf{v_{jk}}$ ($\mathbf{k=1}$ to $\mathbf{N}$) for a particular gene $\mathbf{j}$
- The expression profile for a particular gene $\mathbf{j}$ is:
  $\mathbf{e_j = (v_{j1}, v_{j2}, v_{j3}, \ldots, v_{jN})}$
- A **difference** between two genes $\mathbf{p}$ and $\mathbf{q}$ may be estimated as a $\mathbf{N}$-dimensional "**distance**" metric between $\mathbf{e_p}$ and $\mathbf{e_q}$
- **Euclidean distance:** $\mathbf{d_{pq} = (1/N \sum_{j=1:N} (v_{jp} - v_{jq})^2)^{1/2}}$
- Other distance metrics: correlation coefficient, city-block, etc.
- If distance is scaled to $[0:1]$, then a **similarity** measure: $\mathbf{s_{pq} = 1 - d_{pq}}$

Why Is It Useful to Cluster the Data?

- Clusters represent one way to **putatively identify similar gene expression** across a set of experiment samples
- **Many ways to cluster** the data:
  C.1 Find genes with similar expression
  C.2 K-means clusters where the number of clusters K is fixed
  C.3 Hierarchical clustering where a binary hierarchy is created
  C.*n* Other methods: Self Organizing Memory (SOM), fuzzy clustering, Support Vector Machines (SVM), etc.

Finding Similar Genes

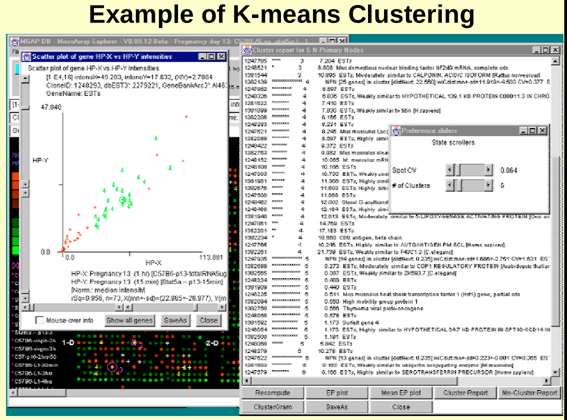
- Find a **sorted list** of all genes $\mathbf{\{g_j\}}$ **similar** to gene $\mathbf{g_s}$
- We define $\mathbf{g_j}$ similar to seed gene $\mathbf{g_s}$ if distance $\mathbf{d_{js}}$ < threshold $\boldsymbol{T}$



K-means Clustering

- **K-means clustering** finds **K** clusters of similar genes. Could use variance of clusters to determine if split into sub-clusters by increasing **K**
- No distance matrix - fast clustering for large numbers of **N** genes
- **Algorithm**:
  1. Pick seed gene **s** and put it into cluster 1 (let **k = 1**)
  2. For all clusters **j = 1 to K**, find gene **q** such that $\mathbf{d_{jq}}$ is a maximum
  3. Set **k = k+1**. Put gene **q** into new cluster **k**
  4. For **j = k to K**, repeat steps 2 and 3 until there are **K** clusters
  5. Then, assign (**N-K**) remaining genes **q** into one of the **K** clusters **j** with minimum $\mathbf{d_{jq}}$
  6. Compute new *virtual* genes as means/medians $\{\mathbf{e_k}\}$ for each of **K** clusters
  7. Reassign all **N** genes **q** into **K** new clusters with minimum $\mathbf{d_{pq}}$ using virtual genes $\{\mathbf{e_p}\}$, compute new $\{\mathbf{e_k}\}$ on new clusters
  8. Variants: use multiple seed genes, range of **K** values, min. COV, medoids

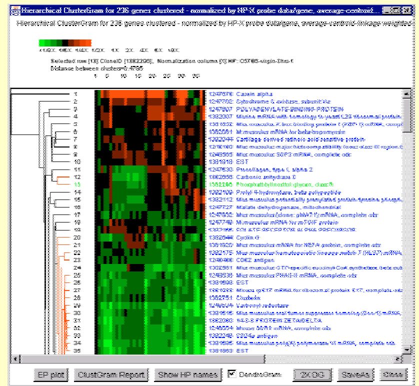
Example of K-means Clustering



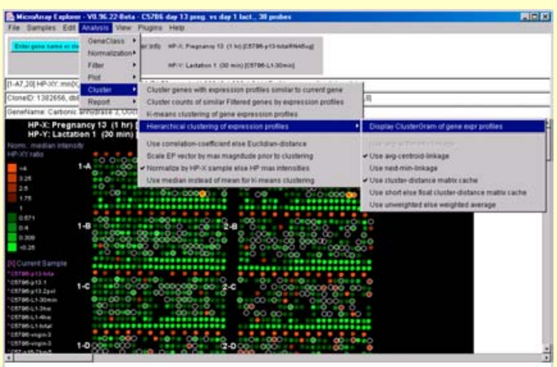
Hierarchical Clustering

- **Hierarchical clustering** requires a distance matrix. For **N** genes (terminal gene clusters), it generates **2N-1** clusters
- **Distance matrix** is a diagonal matrix **D** of d_{jk} of size **N(N-1)/2** (where j and k are genes)
- **Algorithm**:
  1. Assign all **N** genes to terminal clusters 1 to **N**, set **n** to **N**
  2. Find two clusters **p** and **q** such that d_{pq} is a minimum
     - 2.1 Compute a *virtual* cluster vector $\mathbf{e}_{p,q}$ = *average* $(\mathbf{e}_p,\mathbf{e}_q)$
     - 2.2 Set **n = n+1**
     - 2.3 Assign "virtual" cluster to new cluster **n** with estimated value $\mathbf{e}_{p,q}$
  3. Repeat step 2 until **n = 2N-1**

Example of Hierarchical Clustering



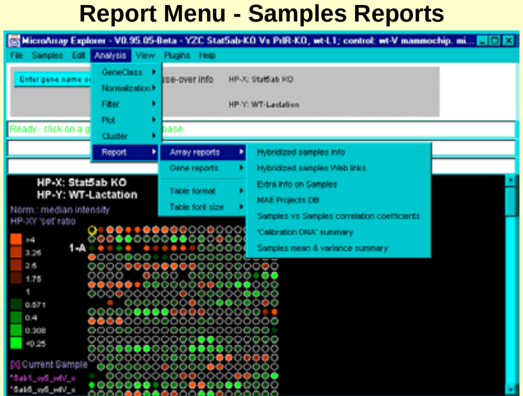
Clustering: Similar Genes, K-means, Hierarchical



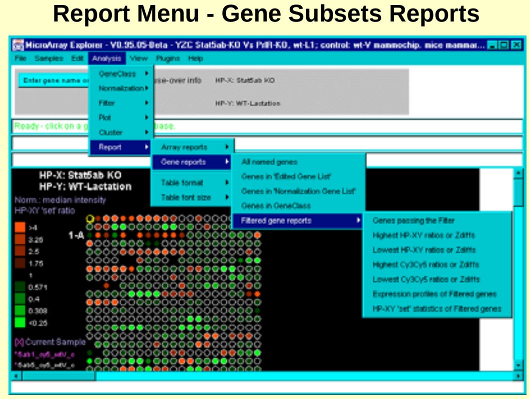
Reports

- **Content:**
  1. Sample Reports: meta-data for samples, correlation coefficients of samples for filtered gene subsets
  2. Gene-reports: gene-reports of particular gene subsets, data filtered genes, highest/lowest expressed genes
- **Formats:**
  1. Dynamic spreadsheets that can access genomic Web databases
  2. Scrollable tab-delimited text - cut & paste to Excel

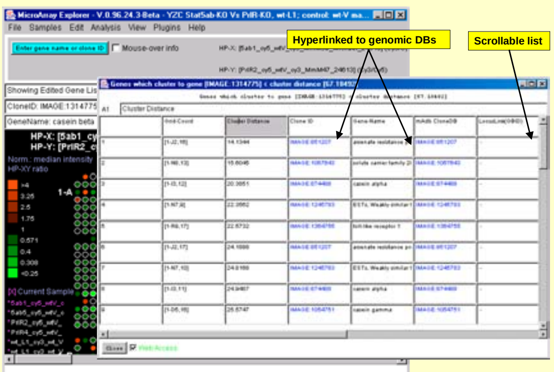
Report Menu - Samples Reports



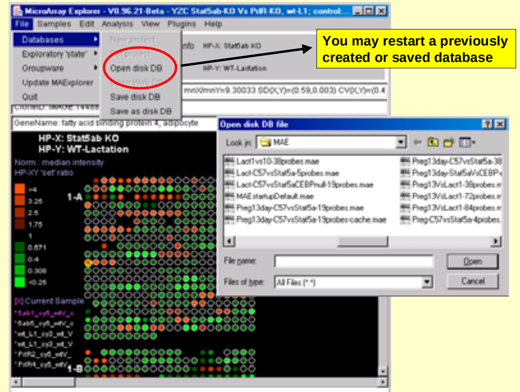
Report Menu - Gene Subsets Reports



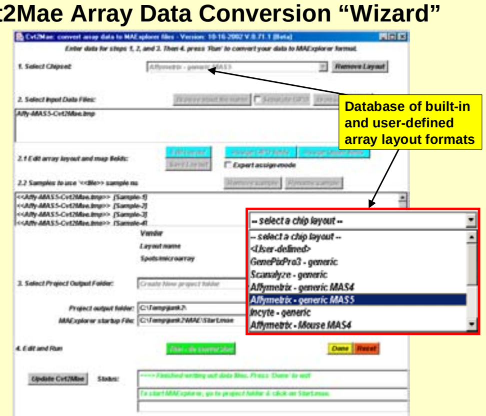
Popup Report of Selected Genes



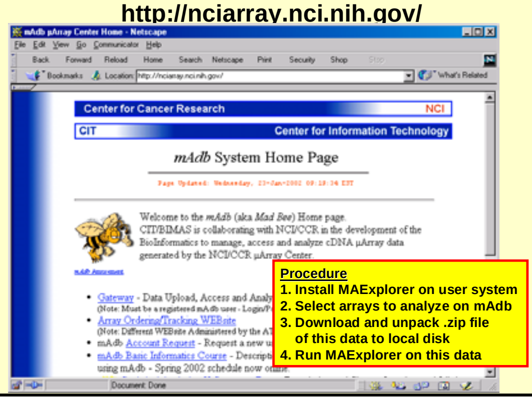
File Menu - Database Access



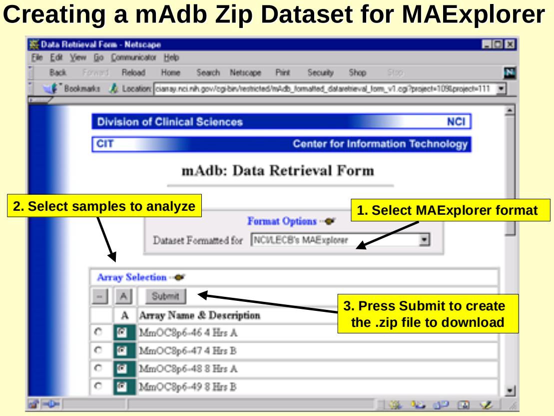
Cvt2Mae Array Data Conversion “Wizard”



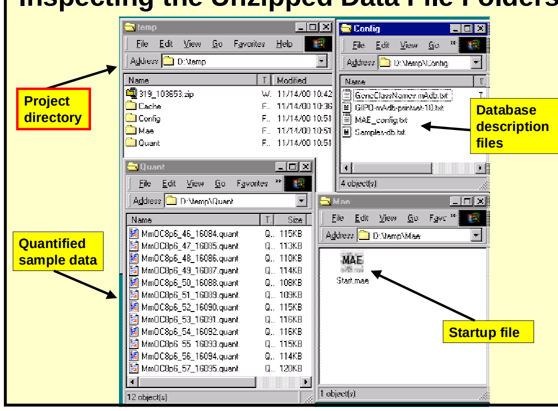
Access Array Data from NCI/CIT mAdb Server http://nciarray.nci.nih.gov/



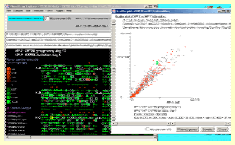
Creating a mAdb Zip Dataset for MAExplorer



Inspecting the Unzipped Data File Folders



Outline



I. Introduction to Data Mining

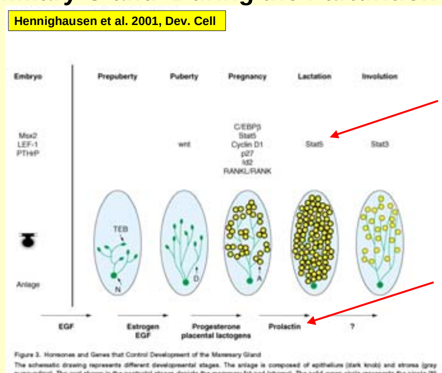
II. Overview of MAExplorer

III. Example

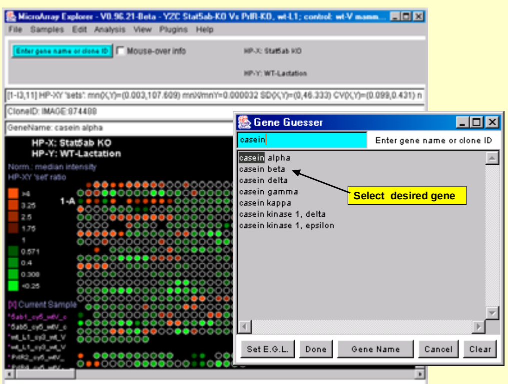
IV. Overview of Java Plug-ins for MAExplorer

V. Overview of Open Source Development

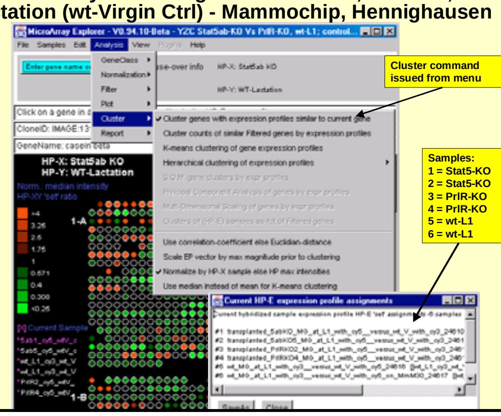
Genes that Control Development in the Mammary Gland During the Parturition Cycle



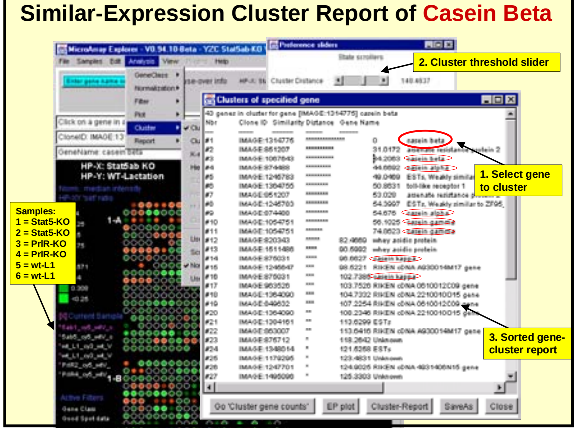
Selecting the Casein Beta Gene By Name



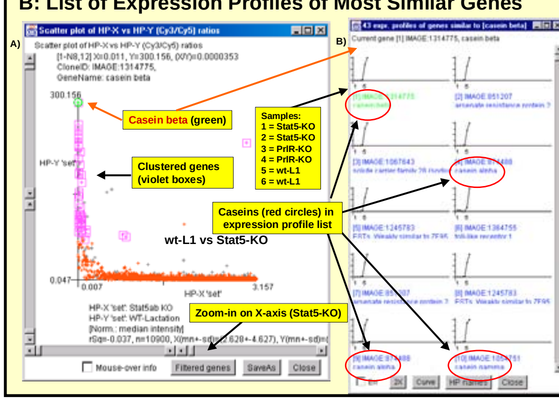
E.g. Similarity Clustering for Stat5ab-KO, PrlR-KO, wt-Lactation (wt-Virgin Ctrl) - Mammochip, Hennighausen



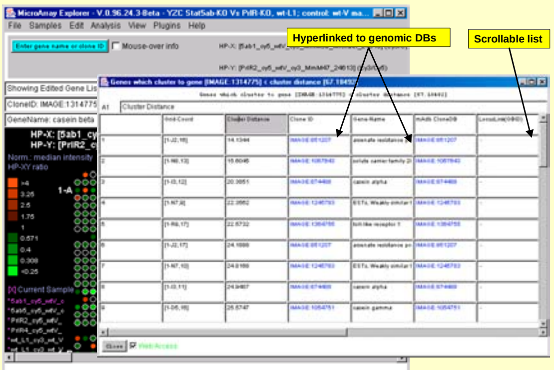
Similar-Expression Cluster Report of Casein Beta



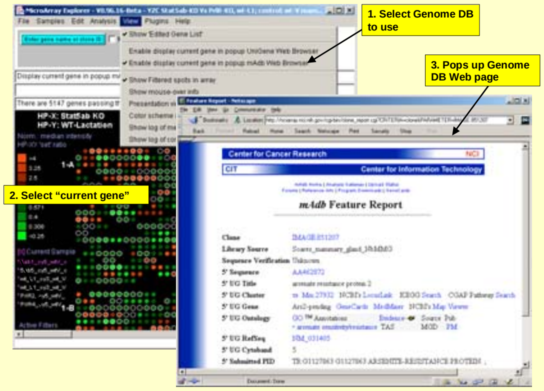
A: Scatter Plot of Genes Similar to Casein Beta
B: List of Expression Profiles of Most Similar Genes



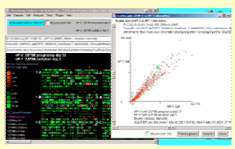
Popup Report of Selected Genes



Popup Web Database Report of Selected Gene



Outline



I. Introduction to Data Mining

II. Overview of MAExplorer

III. Example

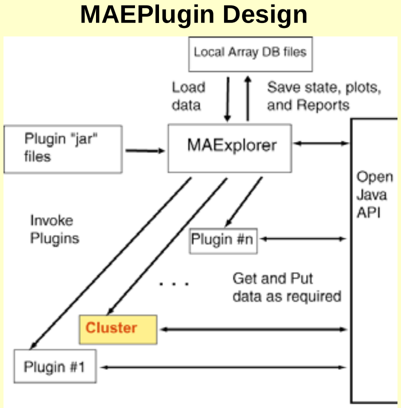
IV. Overview of Java Plug-ins for MAExplorer

V. Overview of Open Source Development

Plug-in Extensions to MAExplorer

- Java plug-ins allow the extension of core MAExplorer program to new analysis methods
- Plug-ins access MAExplorer data structures through the MAExplorer Java Application Programming Interface (API)
- Web site contains: Java API, Java examples, donated plug-ins, and links to plugin-ins
- Plug-ins can provide their own GUI interfaces, may save data back into MAExplorer, or use its plot and report capabilities

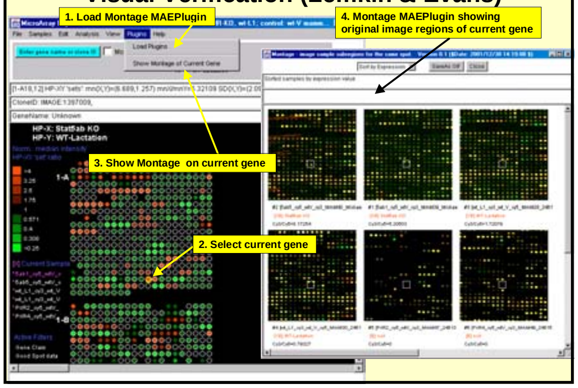
MAEPlugin Design



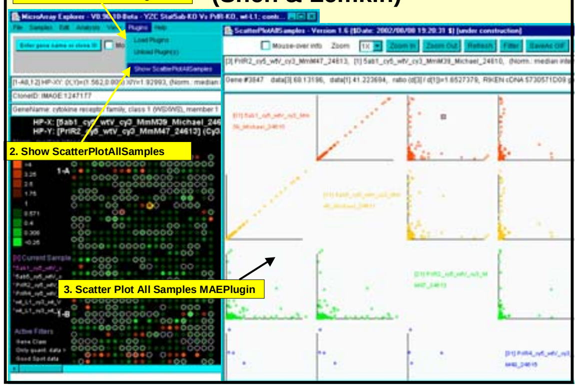
Types of Plug-ins Possible

- Possible MAEPlugin types: normalization, metrics, data filters, PCA, clustering, client-server, functional genomic analysis of cluster results, etc.
- MAEPlugin implementations:
  1. Using 100% Java code
  2. Access local programs written in any language (e.g. 'R' statistics package)
  3. Web-CGI or client-server access to specialized genomic DBs (e.g., GEEVS, MySQL array DB)

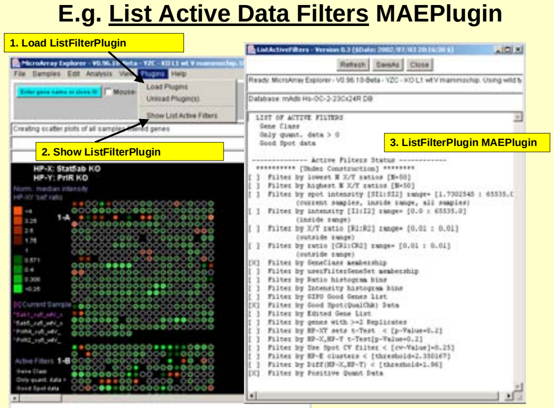
E.g. Montage MAEPlugin of Current Gene for Visual Verification (Lemkin & Evans)



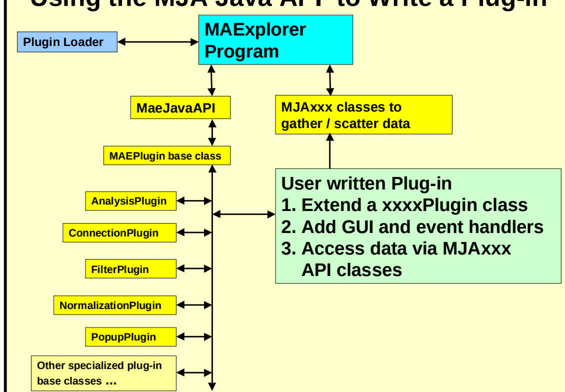
E.g. Scatter Plot MAEPlugin of All Samples (Shen & Lemkin)



E.g. List Active Data Filters MAEPlugin



Using the MJA Java API to Write a Plug-in



MAExplorer Java API Classes Available for User Created MAEPlugins

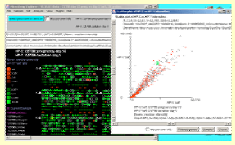
MJAxxxx Class	Objects and method access
MJAbase	base class and constants used by other MJA classes
MJAcluster	cluster data structures and methods
MJAcondition	condition lists of samples and ordered lists of condition lists
MJAeval	command interpreter to invoke MAExplorer commands
MJAexprProfile	expression profiles data
MJAfilter	gene data filters
MJAgene	access single gene genomic and normalized quantified data
MJAgeneList	lists of genes and get sets
MJAgenomicDB	genomic databases on the Internet
MJAgeometry	array geometry, spot to gene maps, etc.
MJAhelp	popup browser help methods
MJAhistogram	histogram plots
MJAmath	built-in math functions
MJAnormalization	normalization data and methods
MJAproperty	get and put individual properties
MJApropList	get lists of properties
MJAsample	get and put single sample lists of spot-level data
MJAsampleList	get lists of samples top-level data
MJAsort	built-in sort methods
MJAstatistics	built-in statistics methods
MJAstate	get and save state, get additional state info
MJAutil	built-in utility methods

Example: "ListFilter" MAEPlugin Java Code++

- The ListFilterPlugin.java class is the class specified to the MAExplorer Java Plugin loader as ListFilterPlugin.jar
  - it installs the menu entry name in MAExplorer
  - it invokes a new instance of ListFilter when selected from the menu
- The ListFilter.java class is called by ListFilterPlugin.java when invoked from the menu
  - it creates a popup GUI extending *Frame*
  - using the MJA classes, it gathers state information on the current MAExplorer data active data filters
  - it draws this information in a *TextArea* in the *Frame*

++ All source code is available on http://maexplorer.sourceforge.net/

Outline



I. Introduction to Data Mining

II. Overview of MAExplorer

III. Example

IV. Overview of Java Plug-ins for MAExplorer

V. Overview of Open Source Development

Open Source - Definition

"The basic idea behind open source is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing."

"We in the open source community have learned that this rapid evolutionary process produces better software than the traditional closed model, in which only a very few programmers can see the source and everybody else must blindly use an opaque block of bits."

The Open Source Initiative (OSI)

http://www.opensource.org/

Open Source - Criteria

- **http://www.opensource.org/docs/definition.php**

1. Free redistribution
2. Source code must be included
3. Derived works must be allowed and may be redistributed
4. Integrity of the author's source code is maintained
5. No discrimination against persons or groups
6. No discrimination against fields of endeavor
7. Distribution of license must be included in all derived works
8. License must not be specific to a product
9. The license must not restrict other software distributed with the software

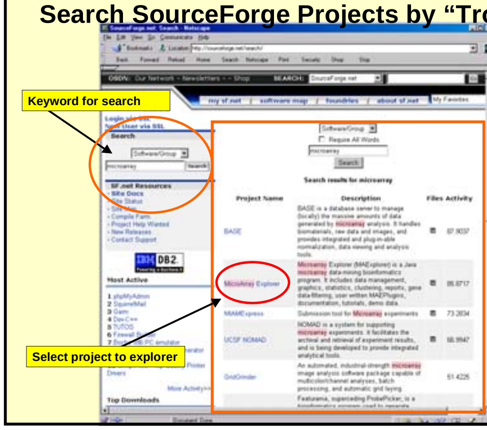
Open Source - Public Licenses

- Approved licenses: **http://www.opensource.org/licenses/**

*Academic Free License, *Apache Software License, *Apple Public Source License, *Artistic license, *Attribution Assurance Licenses, *BSD license, *Common Public License, *Eiffel Forum License, ***GNU General Public License (GPL),** *GNU Library or "Lesser" Public License (LGPL), *IBM Public License, *Intel Open Source License, *Jabber Open Source License, *MIT license, *MITRE Collaborative Virtual Workspace License (CVW License), *Motosoto License, *Mozilla Public License 1.0 (MPL), ***Mozilla Public License 1.1 (MPL),** *Nethack General Public License, *Nokia Open Source License, *Open Group Test Suite License, *Python license (CNRI Python License), *Python Software Foundation License, *Qt Public License (QPL), *Ricoh Source Code Public License, *Sleepycat License, *Sun Industry Standards Source License (SISSL), *Sun Public License, *University of Illinois/NCSA Open Source License, *Vovida Software License v. 1.0, *W3C License, *X.Net License, *Zope Public License, *zlib/libpng license

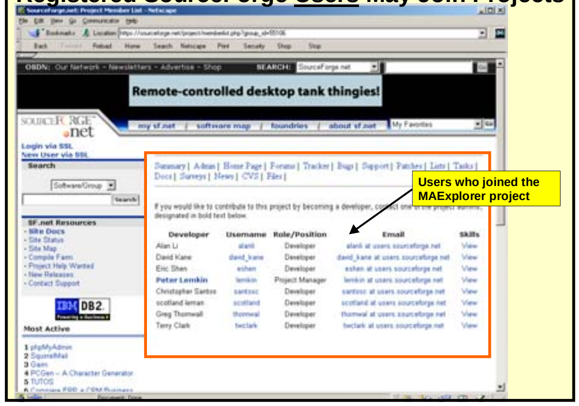
SourceForge.net - Open Source Repository

Search SourceForge Projects by "Trove"

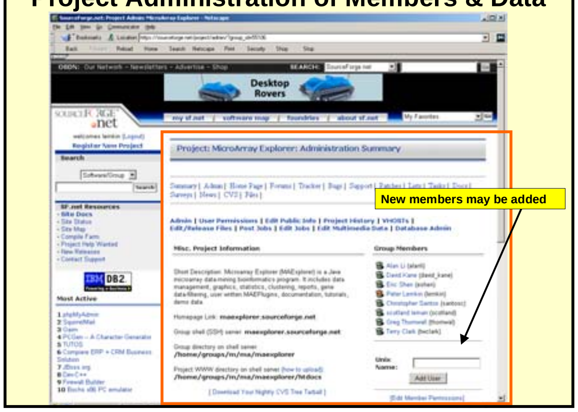


E.g. SourceForge "Project Page"

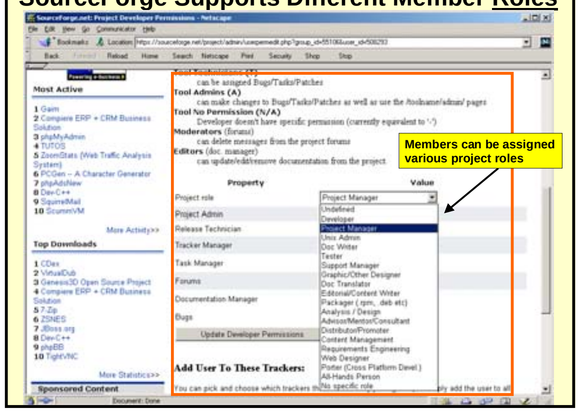
Registered SourceForge Users May Join Projects



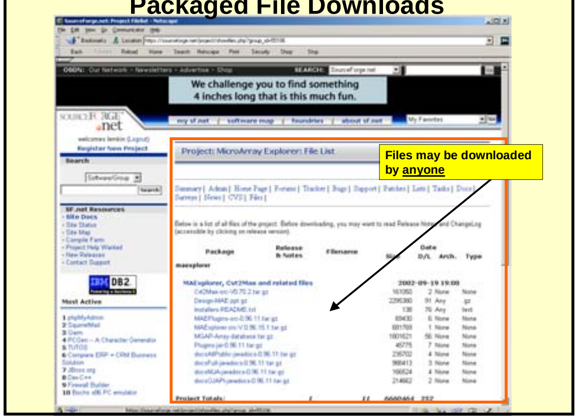
Project Administration of Members & Data



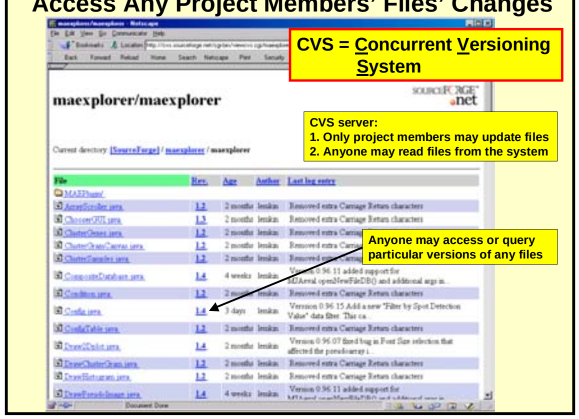
SourceForge Supports Different Member Roles



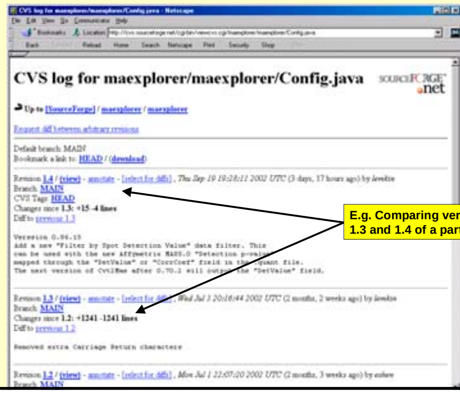
Packaged File Downloads



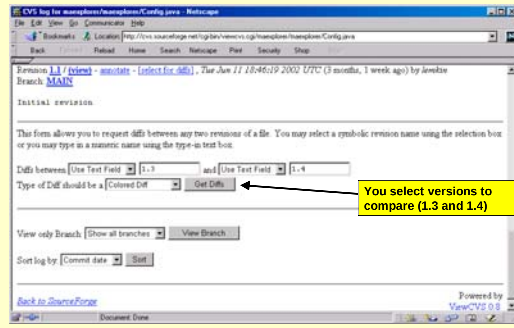
Access Any Project Members' Files' Changes



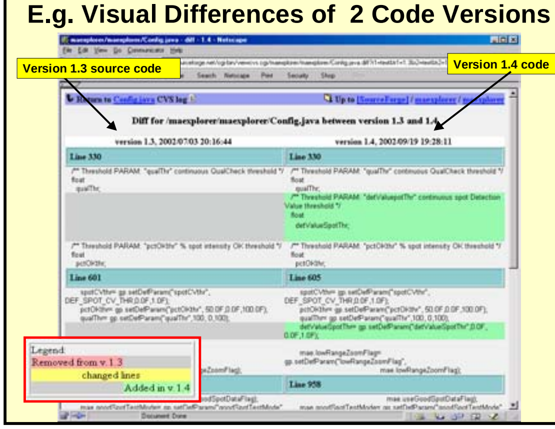
E.g. Comparing Versions of CVS Source Files



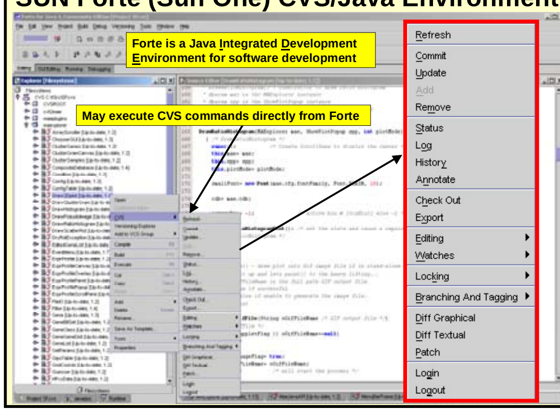
E.g. Select Code Versions to Compare



E.g. Visual Differences of 2 Code Versions



SUN Forte (Sun One) CVS/Java Environment



Some Major Open Source Web Sites

- SourceForge.net - largest open source repository
- Bioinformatics.org - general bioinformatics repository
- Open source projects dedicated to providing language-specific toolkits for processing biological data:
  - BioJava.org
  - BioPerl.org
  - BioPython.org
  - BioXML.org
- BioDAS.org - Distributed Annotation System
- BioCorba.org - data interchange

Summary

- MAExplorer is a flexible fully Open Source microarray data-mining tool
- Uses direct-manipulation, data filtering, built-in graphics, statistics, clustering, gene and sample set operations, reports
- Manages multiple samples, replicates, gene sets, expression profile lists
- Exploration state may be saved and restored
- Accesses genomic Web databases for further analysis
- Convert user tab-delimited data with Cvt2Mae “wizard” tool
- Users may add new analytic methods using Java plug-in extensions

Tutorial Topics - 1

Introduction to array data mining
- microarrays for studying mRNA expression
- overview of cDNA and oligo arrays
- the problem
- things to consider in data mining
- the data mining process

Tutorial Topics - 2

Overview of MAExplorer
- what is MicroArray Explorer
- data preparation
- MAExplorer home page on SourceForge.Net
- documentation and Reference Manual
- downloads and installation on your computer
- user interface and menu system
- sample condition sets and lists - replicates & expression lists
- types of quantified data - intensity or ratio
- simple normalization methods to compare data between samples
- gene data filters
- data management by named gene and sample condition sets
- graphic plots

Tutorial Topics - 2 (continued)

Overview of MAExplorer
- measures of gene similarity
- the gene expression profile
- finding similar genes
- K-means(-median) clustering
- hierarchical clustering
- sample and gene reports
- - hyperlinked spreadsheets that can access genomic Web DBs
- - tab-delimited text for export to Excel
- database access - File menu
- saving and restoring the data-mining session state
- the Cvt2Mae data conversion wizard
- using mAdb data with MAExplorer

Tutorial Topics - 3

Example

- selecting a gene to cluster
- similarity clustering to find a gene subset
- viewing clustered genes in scatter plots, expression profile plots, and reports
- genomic database reports of selected genes

Tutorial Topics - 4

Extending analysis methods using MAExplorer plug-ins (MAEPlugins)
- adding new analysis methods by writing Java Plug-ins
- plug-in design
- dynamic loading of MAEPlugins
- examples of typical plug-ins
- types and implementations of plug-ins
- the MJA Java API (Application Programming Interface)
- top-level example of a MAEPlugin

Tutorial Topics - 5

Open Source
- Definition and criteria
- Public licenses
- SourceForge.Net - Open Source repository
- How an Open Source project works in practice
- Use of CVS in collaborative software development
- Open Source in the bioinformatics area